**The HTTP Protocol**

***Before you start, you should create a document (or a just a piece of paper) where you should write down the Status Code generated by each of the following exercises (You need this for exercise 4-c)***

**1) Monitoring HTTP Headers 1**

**Create a new NetBeans Maven Web-project.**

**For this exercise, we will just use the default index.html generated by NetBeans.**

**Press the run button. When you see the file in the browser (Chrome), open the network tab in the developer window (Ctrl-shift-j) and press F5**

**Observe and explain each of the values monitored (use view source to see the plain messages).**

*Answer:*

*304 – Not Modified*

*This means that the client can continue to use the cached version of the browser.*

*200 – Success*

**Go back to NetBeans and rename your file to index1.html**

**Go back to your browser and (while the developer window is open) change the url to point to the new file.**

**Observe the values**

**Press F5 and observe the values again.**

**Explain what you see.**

I get 200, which means success.

If I refresh I get 200 and 304.

**2) Monitoring HTTP Headers 2**

**Add an image to the page**

**Add an external style sheet to the page <link rel="stylesheet" type="text/css" href="myStyle.css">**

**Reload the page again, observe the request(s) being made, and explain the purpose of the connection header.**

It makes sure that the connection stays open for “KeepAliveTimeout” as long as there is enough “MaxKeepAliveRequests” slots free. Means that you don’t have to handshake each request.

**3) Monitoring HTTP Headers 3  (Response-codes 3xx)**

**In the Web-project, created for 1+2, add a new HTML-page called r.html and add this text in an h1-tag “You got redirected to me”.**

**Use the Wizard to create a servlet called redirect**

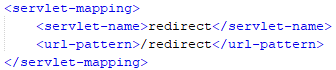
**Remove the processRequest and the doPost method completely from the generated servlet-code.**

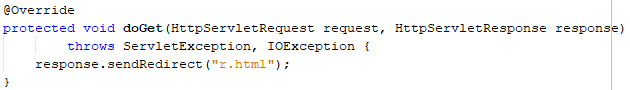
**In the doGet(..) method replace the call to processRequest with this line: response.sendRedirect("r.html");**

**While your server is running, open a (Chrome) browser, and Developer Tools and the network tab.**

**Enter the address for the servlet (http:localhost:8080/redirect) into the browser and explain:**

* **The two HTTP-request you see**
  + 200
  + 302 – Found - This response code means that the URI of requested resource has been changed temporarily. New changes in the URI might be made in the future. Therefore, this same URI should be used by the client in future requests.
* How the browser knew where to go in the second request
  + In the web.xml this is stated:



* + That sends the request to the servlet. 
  + That sends back a redirect response to r.html (200).

**3a) Redirecting to HTTPs instead of HTTP**

**In Chrome enter this address (with the developer window + the network-tab open), and exactly as it is spelt:** [**http://studypoints.info**](http://studypoints.info)

**Explain the first two request monitored (notice where you requested to go, and where you actually ended).**

That address has been moved permanently and gives me a 301 HTTP-Request

Important: Later this week, you will learn how to set up your own server to use https, and ONLY https.

**4a) Status Codes (5xx)**

**Use the Wizard to create a servlet called Ups**

**In the processRequest(..) method,  just before the try-statement add this code:**

**int result = 100/0;**

**While your server is running, open Chrome developer tools and the network tab and then call the servlet.**

**Write down the response code generated by the server as for the previous exercises**

**500 :**

**java.lang.ArithmeticException: / by zero**

Error code 500

**4b) Status Codes (4xx)**

**While your server is running, open Chrome developer tools and the network tab, and call this address: http://localhost:8080/i\_dont\_exist**

**Write down the response code generated by the server as for the previous exercises**

Error code 400

**4c) Status Codes – Ranges**

**Your document, containing the Status Codes for all the exercises done so far, should now contain codes like 2xx, 3xx, 4xx and 5xx.**

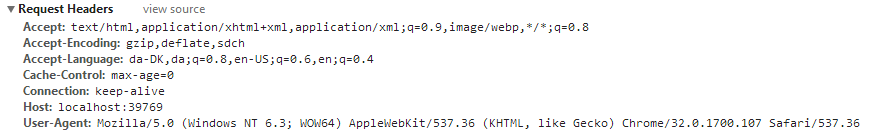
**Explain (write down your answer so you won’t forget) the meaning of the first digit in the 3-digit Status Codes you have seen so far.**

List of codes:

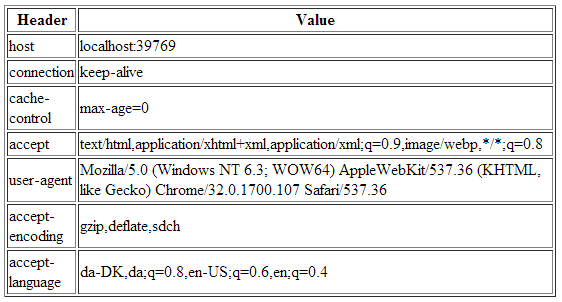
* 404 – Not found
* 500 – Internal Server Error
* 301 – Moved Permanently
* 200 – Success
* 302 – Found
* 304 – Not modified

**5) Get HTTP Request Headers on the Server**

**We have seen that an HTTP request from a Browser typically includes a lot of headers with information related to the client.**

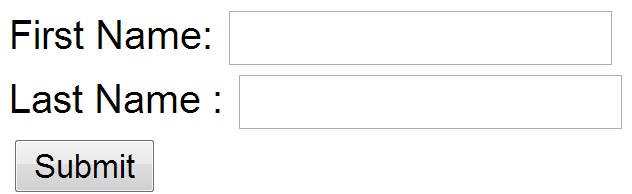


**This information is available to a servlet (actually to any web-server technology)  via the request object. Create a Servlet, which should output this information in a table as sketched in this figure (or in any way you like, but don’t focus on presentation).**



***Hints: Use the request objects getHeaderXXX methods.***

**6) Get/Post-parameters**

**Create a new HTML-file in the web-project made in exercise 1.**

**Add a form to the file, including two text input boxes and a submit button as sketched below:**

**Add an extra input field to the form with type=”hidden”, name=”hidden” and value=12345678.**

**Add the value “#” for the forms action attribute.**

**Set the forms method-attribute to the value “GET” (actually the default value) and test the form. Observe what happens in your browser's address field.**

I get ?hidden=12345678 at the end of the URL

Det her var så langt jeg nåede, det tog sin tid ☺